



Technical University of Lodz
Institute of Electronics

Algorytmy i struktury danych

5. Pamięć komputera

Łódź 2013





Bity i bajty

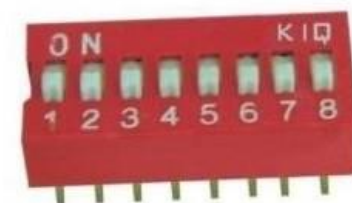
Pamięć komputera jest kategoryzowana wg dostępu, szybkości i pojemności.

Typ	Szybkość dostępu	Odległość do CPU	Pojemność	Ulotna?
Rejestr	Najszybsza	Wewnątrz	10's	tak
Podręczna	B. szybka	Obok	MB	tak
RAM	Szybka	Blisko	GB	tak
Dysk twardy	Wolna	Daleko	TB	nie



- Pamięć komputera można sobie wyobrazić jako ciąg elektronicznych łączników „on/off”. Każdy z nich to **bit (b)**. Grupa 8 bitów to **bajt (B)**.
- Bity są interpretowane jako liczby: „off” to 0, a „on” to 1, np:

off	on	on	off	on	off	off	off
0	1	1	0	1	0	0	0





Liczby dziesiętne i dwójkowe

Liczby, których najczęściej używamy to liczby **dziesiętne** albo o **podstawie 10**.

2	1	7	4	= 2*1000 + 1*100 + 7*10 + 4*1
1000	100	10	1	
10^3	10^2	10^1	10^0	

Liczby **dwójkowe** albo o **podstawie 2** są sumą potęg liczby 2, a nie 10.

1	0	1	1	= 1*8 + 0*4 + 1*2 + 1*1 = 11
8	4	2	1	
2^3	2^2	2^1	2^0	

```
>>> bin(11)
```

```
>>> int(0b1011)
```

```
>>> bin(2174) #Bardzo długi ciąg!
```



Liczby szesnastkowe

Liczby szesnastkowe są liczbami o **podstawie 16**; są sumą potęg liczby 16 mnożonych przez następujące liczby:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Każde cztery bity można uważać za pojedynczą szesnastkową „cyfrę”, ponieważ cztery bity mogą reprezentować liczby z zakresu od 0 do 15.

```
>>> hex(15)
>>> hex(255)
>>> hex(2**32)
>>> hex(-1024)
```

- **Liczby całkowite bez znaku** są przechowywane w pamięci jak liczby dwójkowe.
- **Liczby całkowite ze znakiem** są przechowywane z użyciem **kodu uzupełnień do dwóch**; reprezentacja ta jest poza zakresem niniejszego przedmiotu.



Rozmiar pamięci

- Pojemność pamięci wyraża się w bajtach z prefiksem, dla określenia skali. Prefiksy są zapożyczone z systemu dziesiętnego:

Prefiks	Wartość
kilo-	$10^3 = 1000 = 1$ tysiąc
mega-	$10^6 = 1,000,000 = 1$ milion
giga-	$10^9 = 1,000,000,000 = 1$ miliard
tera-	$10^{12} = 1,000,000,000,000 = 1$ bilion

- Ponieważ pamięć komputera jest zbiorem bitów, do określenia rozmiaru pamięci używa się potęg liczby 2.
- Wartości te są bliskie odpowiednim wielkościom dziesiętnym, ale nie są dokładnie takie same.

Prefiks	Wartość
kilo-	$2^{10} = 1024$
mega-	$2^{20} = 1,048,576$
giga-	$2^{30} = 1,073,741,824$
tera-	$2^{40} = 1,099,511,627,776$



Liczby zmiennoprzecinkowe

Jaki będzie wynik obliczeń poniższego programu?

```
>>> x = 0
>>> while x != 1:
>>>     print (x)
>>>     x += 0.1
```

Uruchom go. Czy otrzymałeś oczekiwany rezultat? Czy 0.1 równa się 1/10?

```
>>> x = 0.1
>>> print x
>>> print „%.32f” % (x)
```

Liczby są reprezentowane **z podstawą dwójkową**; liczba 0.1 ma nieskończony ciąg binarny w tej reprezentacji, $1/10 = 0.0001100110011...$

Podobnie, liczba $1/3$ nie może być zapisana dokładnie w **systemie dziesiętnym**, $1/3 = 0.333333...$



Liczby zmiennoprzecinkowe

Ułamki są przetwarzane na liczby dwójkowe tak samo jak liczby całkowite, tyle że korzysta się przy tym z ujemnych potęg liczby 2, np.

0.	1	0	1	1
	2^{-1}	2^{-2}	2^{-3}	2^{-4}

$$= 1 \cdot 1/2 + 0 \cdot 1/4 + 1 \cdot 1/8 + 1 \cdot 1/16 = 11/16$$

Liczba $1/16$ ma skończoną reprezentację dwójkową, ale nie ma jej liczba $1/10$: $1/10 = 0.0001100110011...$

Z tego powodu liczba $1/10$ nie może być zapisana **dokładnei** jako wartość zmiennoprzecinkowa w komputerze binarnym.

Wniosek: Do porównania liczb zmiennoprzecinkowych używaj **nierówności**.

Ponieważ wielu wartości liczb zmiennoprzecinkowych nie można zapisać dokładnie w pamięci, porównanie tych liczb z użyciem operatorów `==` lub `!=` jest ryzykowne. Gdzie to tylko możliwe, należy stosować operatory z nierównościami (`<`, `<=`, `>`, `>=`).



Ćwiczenia

5.1. Jak pomnożyć liczbę dwójkową przez 2 bez dodawania ani mnożenia? Jak pomnożyć ją przez 8? Podzielić przez 2?

5.2. Jaka największa liczba dwójkowa może być zapisana w jednym bajcie bez znaku? Jaka to wartość w systemie dziesiętnym? Szesnastkowym?

5.3. Jaka największa liczba dwójkowa może być zapisana w słowie dwubajtowym bez znaku? Jaka to wartość w systemie dziesiętnym? Szesnastkowym?

5.4. Określ największą liczbę całkowitą, która może być zapisana w maszynie 32-bitowej.

5.5. Określ największą liczbę całkowitą, która może być zapisana w maszynie 64-bitowej.



Ćwiczenia

- 5.6. Napisz program **binhex.py**, który wyświetla tabelkę wartości binarnych i heksadecymalnych dla dziesiętnych liczb całkowitych od 1 do 36. Wskazówka: Użyj **%ns**, gdzie *n* jest liczbą cyfr, dla sformatowania wyświetlanych wartości.
- 5.7. Zamień następujące wartości dziesiętne na dwójkowe. Które z nich mogą, a które nie mogą być zapisane dokładnie?
a) 0.25, b) 0.375, c) 5.125, d) 10.4375, e) 0.3, f) 0.5
- 5.8. Zamień poniższe wartości dwójkowe na dziesiętne
a) 0.1, b) 0.0101, c) 0.111, d) 10100.01, e) 111.1011, f) 1000.10001
- 5.9. Popraw kod pętli **while** na slajdzie nr 6, tak by obliczenia zakończyły się w oczekiwanej sytuacji.



Podsumowanie

- 1) Liczby binarne są liczbami o podstawie dwa i działają tak samo jak dziesiętne, z takim wyjątkiem, że wykorzystują potęgi liczby 2, a nie 10.
- 2) Podstawą liczb heksadecymalnych jest liczba 16. Zapewniają one krótszy zapis – każde 4 bity są zastąpione przez jeden symbol {0-9,A-F}.
- 3) Jeśli pojemność pamięci zostaje określona z wykorzystaniem potęg liczby dwa, indeksy kilo-, mega- giga-, tera- itd. oznaczają inne wartości niż w przypadku specyfikacji dziesiętnej.
- 4) Komputery wykorzystują dwójkową reprezentację ułamków, w związku z czym wiele wartości zmiennoprzecinkowych, jak 1/100, 10.5 jest zapisanych z błędami.
- 5) Operatory == oraz != nie powinny być używane do porównywania liczb zmiennoprzecinkowych; dla liczb takich odpowiednie są nierówności.



Literatura

Brian Heinold, Introduction to Programming Using Python, Mount St. Mary's University, 2012 (<http://faculty.msmary.edu/heinold/python.html>).

Brad Dayley, Python Phrasebook: Essential Code and Commands, SAMS Publishing, 2007 (dostępne też tłumaczenie: B. Dayley, Python. Rozmówki, Helion, 2007).

Mark J. Johnson, A Concise Introduction to Programming in Python, CRC Press, 2012.